



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
GORTY ET AL.)
) Examiner: J. Hu
Serial No. 10/767,571)
)
Filing Date: January 29, 2004)
)
Confirmation No. 6197) Art Unit: 2154
)
For: SYSTEM AND METHOD OF POLLING)
ELECTRONIC MAILBOXES)
)

DECLARATION UNDER 37 C.F.R. 1.131

Mail Stop Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

We, Suryanarayana Murthy GORTY and Shaibal ROY,
do hereby declare and state:

1. We are the co-inventors of claims 1-33 as
originally filed in the above-identified patent
application.

2. We conceived and reduced to practice the
subject matter of the above-identified patent application
while working in our offices in the United States at TeamOn
Systems, Inc., 1180 NW Maple Street, Issaquah, Washington
98027, prior to August 7, 2003, the effective date of U.S.
patent application publication no. 2005/0039048 to Tosey.

3. We conceived of a communications system and
method of polling electronic mailboxes in which a polling

In re Patent Application of:
GORTY ET AL.
Serial No. 10/767,571
File Date: January 29, 2004

agent polls an electronic mailbox to retrieve unique identifiers (UID's) of electronic messages. A database stores the UID's resulting from the polling. The polling agent is operative for polling the electronic mailbox and retrieving only those UID's that are newer than the UID's from a previous polling to determine that new messages are available.

4. We worked diligently and, as shown in Exhibit 1, had drafted details of a functional specification before August 7, 2003. The functional specification shown in Exhibit 1 indicates the type of details and software engine that will make polling efficient to retrieve UID's from a source, such as in reverse chronological order. Technical details of the optimization for the system and method are also set forth. This functional specification indicates that we had worked out many of the details of the claimed invention to determine the best operating manner. In a few weeks after this functional specification was completed, we reduced to practice and implemented the software for operation.

5. After we had reduced to practice the claimed invention before August 7, 2003, we worked diligently to improve the function in the reduced to practice software and made some code modifications.

6. After working with our patent attorneys, we filed a patent application on the system and method as claimed on January 29, 2004.

7. Exhibit 1 has some redacted dates and potential customer names.

Received at: 7:28PM, 6/13/2005

JUN 13 2005 3:03PM RESEARCH IN MOTION

425-837-8098

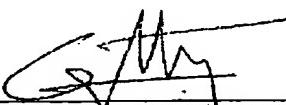
P.2

In re Patent Application of:
GORTY ET AL.
Serial No. 10/767,571
File Date: January 29, 2004

8. We hereby declare that all statements made herein are of my own knowledge and are true and all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title XVIII of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

06/13/2005

Date


Suryanarayana Murthy GORTY

Date

Shaibal ROY

In re Patent Application of:
GORTY ET AL.
Serial No. **10/767,571**
File Date: **January 29, 2004**

8. We hereby declare that all statements made herein are of my own knowledge and are true and all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title XVIII of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date

Suryanarayana Murthy GORTY

June 14, 2005

Date

Shaibal Roy

Shaibal ROY

TeamOn Feature Summary

Title: Polling Optimization

File: Arch_Polling Optimization.doc

Date: February 1, 2003

Author: Jon Smith

| | |
|------------------------------------|---|
| Revision history: | 1 |
| Open issues, questions & comments: | 1 |
| Overview: | 1 |
| Goals: | 1 |
| Current process: | 1 |
| Assumptions: | 1 |
| Scope: | 1 |
| Details: | 2 |

Revision history:

2003 – Jon – Shaibal's original e-mail defining the feature

Open issues, questions & comments:

Overview:

{Provide a one-page paragraph of the feature}

Goals:

[Define the goals of the project and specific objectives within each of the goals.]

Current process:

[Describe the current process, if any.]

Assumptions:

List assumptions made.

Scope:

[Define the scope of the feature, what will and will not be included.]

Details:

[Provide the details of the feature including implications to UI, security, documentation, training, existing processes, existing partners, etc.]

"Polling optimization"

The objectives will include:

- Reduce our Exodus bill (and that of others using MOP)
- Reduce workload on POP mail providers
- Reduce workload on OWA 5.5 servers
- Reduce workload on OWA 2k and iNotes servers

Note that the first two objectives may turn out to be the same, given the preponderance of POP3 among our current customers. The only question here is whether a large enough number of the POP mailboxes that we are polling contain a large number of messages. If so, optimization of the POP3 polls by itself should bring down our Exodus bills. If not, we may need to do POP, OWA 5.5 and OWA 2k to bring down the Exodus bills.

Technical details:

One way to make polling more efficient is to make the vast majority of polls "new-mail-only" polls. Such polls do not check to see whether any messages have been deleted from the source. Only a small minority of polls will check the entire source mailbox to see what has been deleted. An efficient implementation of "new-mail-only" polls will have AggEngine retrieve UIDs from the source in small batches in reverse chronological order (we may have to make some assumptions about the chronological order at the source) until it sees a UID that already exists in the database. AggCron will make the choice between "new-mail-only" and "regular" polls, choosing the former much more frequently than the latter.

More Technical details:

The "new-mail-only" polls don't work very well for our POP Proxy, which needs the list of UIDs in our database to reflect the deletions at the source. This optimization will keep the UIDs in our database longer after the messages have been deleted at the source. However, we can't just get away by making the polls triggered by POP logins to be "regular"-- those are precisely the polls that we need to be fast. Let me elaborate on this in my next email -- but let's not do anything special for POP Proxy in this FPL item.

-shaibal

Darren collected the following data to help us determine where polling needs optimization. Let me pull out a part of it as highlight:

```
QUERY (number of Uids stored for active source mailboxes, by protocol):
protocolName
-----
domino      30995
imap        200
native     100594
owa       162961
pop       728465
rpa        47
```

That tells me that the top four targets for optimization are:

- 1) pop

- 2) owa
- 3) native
- 4) domino

Now, owa is not split off into owa 5.5 and owa 2k. But I think this optimization is more important for 5.5 than for 2k. IF you buy that, the priority goes like this:

- 1) pop
- 2) owa 5.5
- 3) owa 2k
- 4) native (already covered by owa 2k?)
- 5) domino

BTW, there is an iNotes account with over 23,000 uids!

-shaibal

-----Original Message-----

From: D Gardner [mailto:dgardner@hq.teamon.com]
Sent: .. 2003 5:08 PM
To: Shaibal Roy
Subject: Uid numbers

Shaibal, here are the numbers from the ... system

QUERY (number of active source mailboxes, by protocol):

set transaction isolation level 0
go
select smb.protocolName
, count(*)
from SrcMbox smb
where smb.nextPoll <> 9999999999
group by smb.protocolName
go
set transaction isolation level 1
go

RESULT:

protocolName

CS2000 449
domino 12
imap 1228
native 328
owa 431
pop 23509
rpa 4
(7 rows affected)

QUERY (number of Uids stored for active source mailboxes, by protocol):

set transaction isolation level 0

```

go
select smb.protocolName
      ,count(*)
  from SrcMbox smb
     ,SrcMboxMsg smbm
 where smb.nextPoll <> 9999999999
   and smbm.srcMboxID = smb.srcMboxID
 group by smb.protocolName
go
set transaction isolation level 1
go

```

RESULT:

```

-----
protocolName
-----
domino      30995
imap        200
native      100594
owa         162961
pop         728465
rpa          47
(6 rows affected)
```

 THEREFORE, the average number of Uids per active source mailbox, by protocol, sorted ascending by AVERAGE:

| protocolName | NumSrcMboxes | NumUids | AVERAGE | |
|--------------------------|--------------|---------|---------|--|
| CS2000 | 449 | 0 | 0 | |
| imap | 1228 | 200 | 0.16 | <-- Odd. Why do IMAP |
| sources | | | | |
| have Uids? See (1) below | | | | |
| rpa | 4 | 47 | 12 | |
| pop | 23509 | 728465 | 31 | |
| native | 328 | 100594 | 307 | |
| owa | 431 | 162961 | 378 | |
| domino | 12 | 30995 | 2583 | <-- Odd. Very large! See (2) below |

(1) It turns out that there are 6 IMAP source mailboxes with Uids:

| srcMboxID | NumUids |
|-------------------|---------|
| 121901 | 2 |
| 99620 | 4 |
| 155490 | 11 |
| 158244 | 19 |
| 131557 | 21 |
| 122489 | 162 |
| (6 rows affected) | |

I will look into these. Perhaps they converted a POP source into an IMAP

```
source(?)  
  
(2) Since there are only 12 domino source mailboxes, I created a  
histogram,  
which point to one odd mailbox:  
  
srcMboxID          NumUids  
-----  
88585              1  
121236             1  
128721             11  
96063              13  
153409             13  
147878             139  
62318              238  
95250              311  
88894              1436  
143683             1694  
91490              3936  
156273             23205      <-- This one looks pretty goofy. It  
is  
NOT a test account. It was created on  
(12 rows affected)
```

I will try to break down the numbers a bit more. Let me know if you
want me
to run the OnCommand numbers as well.

Darren